Amendments to the Specification:

Please add the following new paragraphs after the end of the paragraph beginning at page 1, line 21, which starts with "Transmission Control Protocol..."

In standard implementations of TCP, transmission of a data stream begins at a transmitter which divides the data stream into segments containing multiple octets. Each segment is then packaged in a packet, which is then sent onto a network for delivery. During transit, intermediate network devices are permitted to break up a packet into fragments and repackage the smaller segment fragments into multiple packets that are then forwarded to the receiver. Since the receiver will then receive multiple packets corresponding to the single original packet sent from the transmitter, there is not necessarily a one-to-one correspondence between packets sent and packets received.

When a packet is received, the receiver sends an acknowledgement signal (ACK) to the transmitter. The ACK acknowledges receipt of the data stream up to a specified octet sequence number. Thus, the ACK normally contains the last octet sequence number of the most recent packet received. However, in cases where a sent packet is split up into multiple packets during transit, the receiver's ACKs of the various fragmented packets (except the last) specify octet sequence numbers that are less than the last octet sequence number in the original sent packet. Thus, these ACKs are not interpreted by the transmitter as acknowledgements of the whole sent packet. In other words, even though these multiple ACKs are derived from a single packet sent from the transmitter, such ACKs are properly interpreted by the transmitter as corresponding to separate fragmented packets received at the receiver. Note, however, that the ACK signals for such split packets have unique octet sequence numbers, and are not mere duplicates. In fact the following section of the TCP specification RFC 793 published by the Internet Engineering Task Force (http://www.ietf.org/) explicitly declares such ACKs as new or acceptable (i.e. non-duplicate) ACKs:

> "In response to sending data the TCP will receive acknowledgments. The following comparisons are needed to process the acknowledgments.

> SND.UNA = oldest unacknowledged sequence number
>
> SND.NXT = next sequence number to be sent
>
> SEG.ACK = acknowledgment from the receiving TCP (next sequence number expected by the receiving TCP)
>
> SEG.SEQ = first sequence number of a segment
>
> SEG.LEN = the number of octets occupied by the data in the segment (counting SYN and FIN)
>
> SEG.SEQ+SEG.LEN-1 = last sequence number of a segment
>
> A new acknowledgment (called an "acceptable ack"), is one for which the inequality below holds:
>
> SND.UNA < SEG.ACK =< SND.NXT"

Note in particular the inequality condition ("less than equal" rather than "equal") in the last equation above. Also note that the "sequence number" referred to is the octet sequence number (and not a packet number), as made clear by the following earlier section of the same document:

> "The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates."